

responding to said global lock being available by performing the steps of:

acquiring said global lock;

performing said call to said critical area of said operating system; and

releasing said global lock.

7. A method as claimed in claim 6 further comprising the step of:

responding to said thread requiring a call to a non-critical area of said operating system by:

performing said call to said non-critical area of said operating system.

8. A method as claimed in claim 7, wherein said operating system includes a micro kernel operating system and:

AI
Cm't
said step of responding to a thread requiring a call to a critical area, includes responding to a thread requiring a call to a critical area of said micro kernel operating system by requesting a global lock;

said step of performing said call to said critical area includes performing said call to said critical area of said micro kernel operating system; and

said step of performing said call to said non-critical area includes performing said call to said non-critical area of said micro kernel operating system.

9. A method as claimed in claim 8, wherein said micro kernel operating system includes a pre-emptable micro kernel operating system, said method further comprising the steps of:

pre-empting any non-critical threads currently executing on said pre-emptable micro kernel operating system prior to said step of acquiring said global lock; and

reinstating said pre-empted threads following said step of releasing said global lock.

10. A method as claimed in claim 9 wherein said step of performing said call to said critical area comprises the steps of:

entering said critical area of said pre-emptable micro kernel operating system;
executing operating system functions as required by said thread;
locking said critical area of said pre-emptable micro kernel operating system; and
exiting said critical area of said pre-emptable micro kernel operating system.

11. A method as claimed in claim 10, further comprising the step of prioritizing execution of threads in accordance with how their respective call latencies will impact real time operation.

12. A method as claimed in claim 10, wherein said operating system includes a real time operating system, and said method further comprises the step of scheduling execution of said threads to be performed by predetermined time deadlines.

AI
Cm't

13. A computer system comprising:
one or more processors;
a memory medium storing an operating system having critical and non-critical areas, in a machine executable form, and a lock manager in a machine executable form;
a communication network interconnecting said one or more processors, and said memory;
and
said lock manager being operable to:
 respond to a thread requiring a call to a critical area of said operating system by requesting a global lock; and
 respond to said global lock being available by performing the steps of:
 acquiring said global lock;
 performing said call to said critical area of said operating system; and
 releasing said global lock.

14. An apparatus for symmetric multiprocessing comprising:
operating system means having critical and non-critical areas;
means responsive to a thread requiring a call to a critical area of said operating system by
requesting a global lock; and
means responsive to said global lock being available by performing the steps of:
acquiring said global lock;
performing said call to said critical area of said operating system; and
releasing said global lock.

AI
could
15. A computer readable memory medium, storing computer software code executable to
perform the steps of:

responding to a thread requiring a call to a critical area of an operating system having
critical and non-critical areas, by requesting a global lock; and
responding to said global lock being available by performing the steps of:
acquiring said global lock;
performing said call to said critical area of said operating system; and
releasing said global lock.

16. A computer data signal embodied in a carrier wave, said computer data signal
comprising a set of machine executable code being executable by a computer to perform the steps
of:

responding to a thread requiring a call to a critical area of an operating system having
critical and non-critical areas by requesting a global lock; and
responding to said global lock being available by performing the steps of:
acquiring said global lock;
performing said call to said critical area of said operating system; and
releasing said global lock.--